# Sampler Design for Bayesian Personalized Ranking by Leveraging View Data

Jingtao Ding, Guanghui Yu, Xiangnan He, Fuli Feng, Yong Li, *Senior Member, IEEE,* and Depeng Jin, *Member, IEEE*

**Abstract**—Bayesian Personalized Ranking (BPR) is a representative pairwise learning method for optimizing recommendation models. It is widely known that the performance of BPR depends largely on the quality of negative sampler. In this paper, we make two contributions with respect to BPR. First, we find that sampling negative items from the whole space is unnecessary and may even degrade the performance. Second, focusing on the purchase feedback of E-commerce, we propose a negative sampler for BPR by leveraging the additional view data. In our proposed sampler, users' viewed interactions are considered as an intermediate feedback between the purchased and unobserved interactions. We jointly learn the pairwise rankings of user preference among these three types of interactions and design a user-oriented weighting strategy during learning process, which is more effective and flexible. Compared to the vanilla BPR that applies a uniform sampler on all candidates, our view-enhanced sampler enhances BPR with a relative improvement over $36.64\%$ and $16.40\%$ on Beibei and Tmall datasets, respectively. Empirical studies demonstrate the importance of considering users' additional feedback when modeling their preference on different items, which can effectively improve the quality of sampled negative items towards learning a better personalized ranking function. Our implementation is available at https://github.com/dingjingtao/NegativeSamplerBPR.

**Index Terms**—Bayesian personalized ranking; recommendation; sampler; view data.

## 1 Introduction

Due to the prevalence of user implicit feedback in online information systems, recent research on recommendation has shifted from explicit ratings to implicit feedback, such as purchases, clicks, watches and so on [2], [11]. Different from the recommendation with explicit ratings [14], [15], negative feedback is naturally scarce when dealing with implicit feedback, also known as one-class problem [25]. To learn recommender models from binary implicit feedback, Rendle et al. [29] proposed the Bayesian Personalized Ranking (BPR) method, which assumes that an observed interaction should be predicted with a higher score than its unobserved counterparts (*i.e.*, the missing interactions). The optimization of BPR is usually achieved by the stochastic gradient descent (SGD). In each step, it first randomly draws an observed interaction $(u, i)$, and then selects an item $j$ that $u$ has not interacted with before to constitute $(u, i, j)$. Such a process of selecting $j$ is also known as *negative sampling*.

In the original paper of BPR [29], Rendle *et al.* applied a uniform negative sampler, *i.e.*, sampling $j$ from **all items** that $u$ has not consumed before with an **equal probability**. Later on, it was reported that such a uniform negative sampler is highly ineffective and slows down the convergence of BPR [28], [40], especially for datasets that have a large number of items. To this end, Rendle *et al.* [40] proposed dynamic negative sampling (DNS) strategies, aiming to maximize the utility of a gradient step by choosing "difficult" negative examples — *i.e.*, the negative examples that lead to a large prediction loss by the current model. This process is first randomly selecting $X$ candidates and then drawing a "difficult" negative sample with a multinomial distribution based on their prediction scores, where the one with the higher score, *i.e.*, the higher prediction loss, is more likely to be selected. Following this idea of DNS strategy, Rendle *et al.* [28] further proposed a context-dependent sampler that oversamples informative pairs in each step, and developed an efficient implementation with constant amortized runtime costs. Despite the significant improvements have been observed, existing DNS strategies sample negative items from the whole item space, which arguably may still suffer from low efficiency when the number of items is large.

To further mitigate the one-class problem, one intuition is to leverage more side information for learning a more precise preference between two items. In today's implicit recommender systems, besides the primary feedback that can be directly utilized to optimize the conversion rate (CVR), other additional feedback is readily available [8], [33]. Like in E-commerce systems, users' multiple micro-behaviors including view, purchase, wish and put-in-cart are collected [43]. Similarly, there are heterogeneous signals related to users' search and watch history in online video streaming systems [6]. Compared to the primary one, the additional feedback always reflects a relative lower level of preference, which could help in learning user preference. For example, in E-commerce systems, user usually views an

- *J. Ding, G. Yu, Y. Li and D. Jin are with Beijing National Research Center for Information Science and Technology (BNRist) and with Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. E-mail: {dingjt15, ygh15}@mails.tsinghua.edu.cn, {liyong07, jindp}@tsinghua.edu.cn.*
- *X. He is with the University of Science and Technology of China, Hefei, Anhui, China, 230031. E-mail: xiangnanhe@gmail.com.*
- *F. Feng are with School of Computing, National University of Singapore, Singapore. E-mail: fulifeng93@gmail.com*

item before purchasing it. Even though a viewed item is not purchased afterwards, it should still be treated differently when compared with other missing items. Also, searching a specific video in online video streaming systems can also be considered as a relatively weak signal of user preference. As the BPR learns a pairwise ranking relation of user preference between two items, the above additional information can be seamlessly integrated into it by designing an improved BPR sampler.

In this work, we aim to answer the following two research questions: 1) is it necessary to sample negative items from the whole space? and 2) can we design a better sampler for BPR? For the first question about inefficient sampling from whole negative item space, we propose to sample negative items from a reduced space, given that one user normally interacts with a few items. More specifically, we first design a sampler that pre-selects the candidate itemset for each user by uniformly drawing a fixed number of instances from the unobserved items. Though with simplicity, this sampler may suffer from the distortion of probability that each item get selected as the negative instance. Therefore, we further design an improved sampler that manages to reduce the sampling space with negative sampling probability of each item approximately unchanged. As for the second question, focusing on a specific domain of online-shopping recommender systems, we propose a view-enhanced BPR sampler that considers users' viewed interactions as an intermediate feedback between purchased and unobserved (*i.e.*, neither purchased nor viewed) interactions. We first design a biased sampling process that assumes two-fold semantics in a viewed item, *i.e.*, a negative signal when it was sampled together with another purchased item and a positive signal when with another unobserved item. By tuning the corresponding probability in this biased sampling, the trade-off between these two semantics of user's view signal can be achieved. Then, we improve the above scheme by learning the three pairwise ranking relations among a purchased item, a viewed item and an unobserved item together in each training example. In particular, we design a novel objective function with weighted loss to encode the above three relations in the BPR sampler. We further assign the weight of these relations based on users' habits in online-shopping activities, which is arguably more effective than the previous methods [17], [21], [29] that are limited by the uniformity assumption.

We summarize our key contributions of this work as follows.

1. We propose to sample negative items from a reduced item space in BPR and empirically demonstrate that it is unnecessary to sample from all items. When the space is reduced to $1/2^{10}$ of original size, it achieves a relative improvement of at most $1.78\%$ on a popularity-skewed Beibei dataset. And on another less skewed Tmall dataset, it still achieves performance improvement when the reduced space is larger than $1/2^4$ of original size.
2. We design a view-enhanced user-oriented BPR sampler that can effectively integrate users' view data in online-shopping recommender systems, where the viewed interactions are considered as an intermediate feedback between those purchased and unobserved interactions.
3. We conduct extensive experiments on two real-world datasets, showing that our view-enhanced sampler enhances BPR with a relative improvement of $36.64\%$ and $16.40\%$. Furthermore, it outperforms state-of-the-art methods by a large margin, about $2.1\% \sim 9.95\%$.

The rest of this paper is organised as follows. We review related literature in Section 2. Then, we introduce the dataset and experimental settings in Section 3. The two research questions are investigated in Section 4 and Section 5, respectively. Finally, we conclude this work and discuss future work in Section 6.

## 2 Related Work

As implicit feedback data is more common and valuable in modern recommender systems, we first review some related works on modeling user preference from implicit data. Then, we discuss two types of methods that are proposed to improve implicit recommender systems with multiple feedback.

**Implicit Feedback Systems.** Handling missing data is notoriously difficult for recommendation with implicit feedback. To solve this problem, two strategies are proposed: whole-data based strategy and sample-based strategy. Whole-data based strategy treats all missing data as negative feedback [11], [12], [13], while sample-based learning strategy overcomes this problem by sampling negative instances from missing data [25], [29]. Both methods have pros and cons: whole-based methods model the full data with a potentially higher coverage, but inefficiency can be an issue; sample-based methods are more efficient by reducing negative examples in training, but risk decreasing the model's performance. Among the sample-based learning-to-rank methods, many forms of loss functions have been investigated, like squared loss [7] and BPR loss, and the most well-known one is BPR. With the ease of integrating any form of loss function, many neural network based models, like Ref. [11] and [22], adopt the pairwise ranking loss of BPR. Recently, Song *et al.* [31] further proposed a general ranking neural network that includes BPR as a special case. Moreover, this idea of pairwise ranking has also been introduced into community question answering systems [24] and clothing matching areas [32]. Therefore, in this paper, we focus on developing an improved sampler for BPR. Different from previous works, we demonstrate that 1) it is unnecessary to sample negative items from the whole space, and 2) recommendation performance can be significantly improved after integrating users' additional view data.

**Collective Matrix Factorization (CMF).** CMF is a multiple relational learning method that improves predictive accuracy by sharing information between different feedback [4], [30], [39]. Originating from the explicit rating problems, it has been extended into implicit case as well [3], [16], [18], [38], [41]. For example, by applying CMF technique to Bayesian Personalized Ranking, Multi-Relational Factorization with BPR (MR-BPR) performs better on social network data [16]. A recently proposed method [18], namely Multiple Feedback Personalized Ranking (MFPR), borrows the idea of SVD++ [14] to integrate additional feedback and later optimizes a pairwise ranking loss, which is similar to BPR. However, as the CMF-based model generates different user-item relations, *i.e.*, latent factors, for each type of feedback, it is hard to differentiate their preference levels. In contrast, our view-enhanced BPR sampler learns the same user-item relation to indicate relative preference order among purchase and view data, which is more effective.

**BPR-based Models.** The second category of methods integrate multiple types of feedback in the sampler of BPR [17], [21], [26]. The time-based and interaction-count based variants of samplers are designed to provide more signals [17]. From the perspective of transferring knowledge from additional feedback, Pan *et al.* [26] propose an adaptive BPR that integrates these feedback to learn better confidence of users' preference on items. Qiu *et al.* [27] analyze the co-occurrence of different types of actions, based on which the user preference can be learned. Recently, Multi-channel BPR (MC-BPR) applies the strategy of assigning different preference levels to multiple types of feedback when sampling training item pairs in BPR [21], which is similar to our proposed view-enhanced scheme based on a biased sampling process. However, by simultaneously modeling pairwise ranking relations among user's purchased, viewed and unobserved items in each training example, our proposed scheme achieves better performance. Moreover, with a user-oriented weighting scheme, the performance can be further improved.

# 3 Datasets and Observations

## 3.1 Datasets

We perform experiments on two real-world datasets.

**Beibei**[1]: Beibei is the largest E-commerce platform for maternal and infant products in China. We sample a subset of user interactions that contain views and purchases from Beibei within the time period from 2017/05/25 to 2017/06/28.

**Tmall**[2]: Tmall is the largest business-to-consumer E-commerce platform in China. To allow our results to be reproducible, we use a public benchmark released by the IJCAI-2015[3]. The time period is from 2014/06/01 to 2014/11/11. Note that $11^{th}$ Nov. of each year is the Tmall Global Shopping Festival[4], and thus users tend to select many items before and wait for the deals on this day. Therefore, in order to filter out the possible effect brought by this shopping festival, we drop the interactions after October and obtain a subset, denoted as **Tmall-select**. As the timestamps in this dataset are at least 40 days before the shopping festival, it is unaffected and much less noisy in terms of user behaviors. We further discuss the validity and limitation of this filtered Tmall-select dataset in Sec. 3.5.

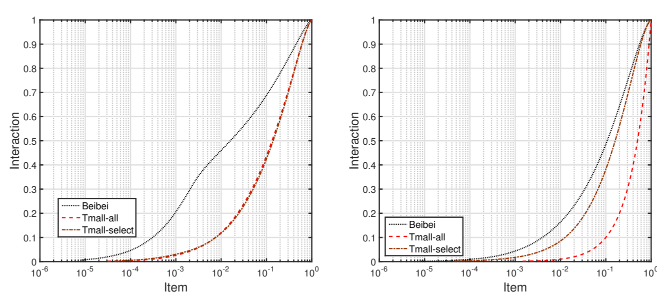
We take three steps for data preprocessing. We first merge the repetitive purchases of the same user and item into one purchase with the earliest timestamp, as we aim to recommend novel items. Next we filter out users' views on their purchased items to avoid information leaking. Finally, we filter out users and items with less than 12 and 16 purchases, respectively, to overcome the high sparsity of the raw datasets. Table 1 summarizes the statistics of our experiment datasets. With both primary (purchase) and additional (view) feedback collected, these datasets are sufficient for our research on leverage additional view data in BPR sampler.

TABLE 1
Statistics of the evaluation datasets.

| Dataset | Purchase# | View# | User# | Item# | Sparsity |
|---|---|---|---|---|---|
| Beibei | 2,654,467 | 23,668,454 | 158,907 | 119,012 | 99.99%/99.87% |
| Tmall-all | 352,768 | 1,585,225 | 28,059 | 32,339 | 99.96%/99.83% |
| Tmall-select | 160,840 | 531,640 | 12,921 | 22,570 | 99.94%/99.82% |

## 3.2 Observations

The popularity skewness exists in many recommender systems and impacts the performance. Therefore, we investigate the popularity skewness in our data, in terms of item purchases and views, and show the result in Fig. 1(a) and (b), respectively. The y-axis represents the ratio of interactions for a given ratio of items on the x-axis, sorted by decreasing popularity. For item purchases, Beibei is the most popularity skewed dataset, where the top-1% of the items accounts for 50% of the purchased interactions, much larger than 10% in Tmall dataset. Such difference in skewness no longer exists in item views, where the top-1% of the items accounts for 16% and 9% of the viewed interactions in Beibei and Tmall-select, respectively. As for the difference between Tmall-all and Tmall-select, the popularity skewness of purchase interactions is almost the same, as shown in Fig. 1(a), while for viewed interactions Tmall-select is much more skewed than Tmall-all, about 40% vs. 10% in terms of top-10% of the items. In summary, users in Beibei are more likely to purchase those popular items, which may affect the performance of personalized recommendation algorithms. On the contrary, users in Tmall-all do not tend to view those popular items, meaning that there may exist a strong personal preference in users' views.




(a) purchased interactions (b) viewed interactions

Fig. 1. Popularity skewness of the Beibei and Tmall datasets.

## 3.3 BPR

The objective function for BPR can be formulated as

$$\arg\min_{\Theta} \sum_{(u,i,j)\in\mathcal{D}} -\ln\sigma(\hat{y}_{ui}(\Theta) - \hat{y}_{uj}(\Theta)), \quad (1)$$

1. http://www.beibei.com/
2. https://www.tmall.com/
3. The dataset is downloaded from https://tianchi.aliyun.com/datalab/dataSet.htm?id=5
4. http://www.alizila.com/look-back-2014-global-shopping-festival/

where $\hat{y}(\Theta)$ is the predictive model, and we use the standard matrix factorization [29] as the predictive model. $\Theta$ denotes the model parameters, $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function to convert the margin to a probability, and $\mathcal{D}$ denotes the set of pairwise training examples: $\{(u,i,j)|i \in \mathcal{R}_u^+ \wedge j \notin \mathcal{R}_u^+\}$, where $\mathcal{R}_u^+$ denotes the set of items that $u$ has interacted with before. Note that we have omitted the $L_2$ regularization terms for clarity. The optimization of BPR is usually achieved by the stochastic gradient descent (SGD).

### 3.4 Evaluation Methodology

We adopt the *leave-one-out* protocol [11], [29], where the latest purchase interaction of each user is held out for testing. For hyperparameter tuning, we randomly sample one purchase interaction for each user as the validation set. The training process is stopped once we observe increasing in the validation loss.

For evaluation measures, we employ *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG). Mathematically, $HR@k$ for each user $u$ is defined as:

$$HR_u@k = \begin{cases} 1\ , & \text{hit in top-k recommendation} \\ 0\ , & \text{else.} \end{cases} \tag{2}$$

$NDCG@k$ for each user $u$ is defined as:

$$NDCG_u@k = \sum_{p=1}^{k} \frac{2^{R(u,p)} - 1}{\log(p+1)}, \tag{3}$$

where $R(u,p)$ is the rating assigned by $u$ to the item at the $p^{th}$ position on the ranked list produced for $u$. Here $R(u,p)$ equals 1 if hit and 0 otherwise. Compared to HR, NDCG is very sensitive to the ratings of the highest ranked items. We truncate the ranked list of non-purchased items at the position of 100, *i.e.*, k=100, and report the average score of all users. We test the learning rate of $[0.0005, 0.001, 0.005, 0.01, 0.05]$. For regularization, we set $\lambda$ as $0.01$ (Beibei) and $0.1$ (Tmall) for all methods for a fair comparison. Since the findings are consistent across the number of latent factors $K$, we report the results of $K = 32$ only.

### 3.5 Discussion

In order to obtain a Tmall-select dataset where users' online behaviors are not affected by shopping festival on Nov. $11^{th}$, we only select interactions that happened at least 40 days before. The threshold of 40 days is set based on the following three observations. First, the users' purchase intentions cannot be affected until the release time of discount information during shopping festival, which is generally two-three weeks before Nov. $11^{th}$ according to some related materials[5,6]. Second, it has been found out that purchase signals in online behaviors are amplied in the last three days before purchase, based on an analysis of over two million Pinterest users purchasing behavior [20]. Last but not least, users that are sensitive to discount promotion only occupy a part of total purchaser base, as Liu *et al.* [19] classified Chinese online purchasers into six types by cluster analysis and analyzed their different sensitivity to promotion strategies. Therefore, based on above three observations, we believe the length of 40 days (*i.e.*, 6 weeks) is long enough to remove the possible effect of online shopping festivals on users' viewing and purchasing behaviors.

However, above solution for the setting of threshold does not consider the factor of different item categories on users' planning buying behaviors [5]. For example, users are more likely to plan the purchases of home appliances several weeks before shopping festivals compared with those of other small items like books. In this case, the filtering threshold for home appliances should be larger. However, the explicit item information in raw data has already been encoded into meaningless ids, which makes it impossible to know what item or which type the users have purchased. Therefore, we are not able to propose a more suitable setting of threshold, which may cause the biased estimation of performance gain from users' view data if most users view target items a long time before the festival. In terms of related works on this field, Zheng et al. [42] observed that a suitable promotion scheme can increase both planned buying and impulse buying in online shopping festivals. Other works like [1] and [37] pointed out that informational incentives (*e.g.* promotional information and review information) and social influence (*e.g.* peer imitation and endorsement influence) are two main positive factors in facilitating consumer behavior during online shopping festivals. However, these two should not affect consumers' behaviors at almost one month before the festival, when the promotion campaign does not begin and the social influence cannot take effect. Therefore, by setting the threshold as 40 days, we are able to remove the possible effect of online shopping festivals on users' purchasing and viewing behaviors in Tmall dataset.

## 4 Unnecessary to Sample from All Items

Generally the vanilla BPR samples negative items indiscriminately from the whole set of those unobserved instances. As the negative sampling space of BPR is fairly large for each user in implicit recommender systems, it may not only cause inefficiency issue but also degrade the performance. To overcome this, we design the following scheme of reducing negative sampling space to evaluate whether it is necessary to sample from all items.

### 4.1 Methodology

In our designed scheme, as detailed in Algorithm 1, each user's negative training instances are randomly sampled from a pre-selected subset of whole item space, which is much smaller but different among the users. More specifically, given the size ratio $\gamma$ of this reduced space to the original space, first we randomly assign each user $\gamma \times N$ samples as the negative sampling space $\mathcal{R}_u^-$ (Lines:1-4). Then, with $\mathcal{R}_u^-$ fixed, BPR sampler randomly draws training samples $(u,i,j)$ and updates model parameters in each iteration (Lines:7-11). Since the negative instances, *i.e.*, $j$, can only be sampled from $\mathcal{R}_u^-$, this scheme reduces the number of possible training item pairs $\{(i,j)\}$ for $u$, and thus can largely improve efficiency in terms of learning model parameters.

5. https://www.chinainternetwatch.com/tag/double-11/
6. https://sea.mashable.com/culture/924/alibaba-tells-us-all-their-pre-1111-promos-and-were-starting

**Algorithm 1:** Proposed scheme of reducing negative sampling space in BPR.

**Input :** number of users $M$ and items $N$, user-item interaction data $\mathcal{S}$, reduce ratio $\gamma$
**Output:** $\Theta$
**for** $u \leftarrow 1$ *to* $M$ **do**
  //Generate the negative sampling space for each user
  $\mathcal{R}_u \leftarrow random\_select(u, N, \gamma)$
**end**
**while** *not reaching convergence* **do**
  // Random sampling
  $u \leftarrow$ draw a random user from $\mathcal{U}$
  $i \leftarrow$ draw a random purchased item from $\mathcal{S}_u$
  $j \leftarrow$ draw a random negative item from $\mathcal{R}_u$
  Compute gradients of $\Theta$ according to BPR
  Update the above parameters
**end**

**Algorithm 2:** Generating reduced item space with the pre-computed item probability.

**Input :** number of users $M$ and items $N$, user-item interaction data $\mathcal{S}$, reduce ratio $\gamma$, negative sampling probability distribution $P(J)$
**Output:** reduced item space $\{\mathcal{R}_u^-\}$
1 //Sample the item candidates $\mathcal{C}$ for all users' $\mathcal{R}_u^-$
2 $\mathcal{C} \leftarrow \{\}$
3 **for** $k \leftarrow 1$ *to* $M \times N \times \gamma$ **do**
4   Draw $j \sim P(J)$
5   $\mathcal{C}.add(j)$
6 **end**
7 **for** $u \leftarrow 1$ *to* $M$ **do**
8   //Generate the negative sampling space for $u$
9   $\mathcal{R}_u^- \leftarrow \{\}$
10   $j \leftarrow \mathcal{C}.first()$
11   **while** $|\mathcal{R}_u^-| < N \times \gamma$ **do**
12     **if** $j \notin \mathcal{S}_u$ **then**
13       $\mathcal{R}_u^-.add(j)$
14       $\mathcal{C}.remove(j)$
15     **end**
16     $j \leftarrow \mathcal{C}.next()$
17   **end**
18 **end**

As for the approach to the $\mathcal{R}_u^-$ generation, an intuitive implementation is to uniformly draw $\gamma \times N$ unobserved instances. Given $u$'s interaction history $\mathcal{S}_u$, $\gamma \times N$ instances are sampled according to a uniform distribution. Compared to the original sampling space, this reduced space introduces an extra bias on unpopular items, increasing the probability in negative sampling, which we will demonstrate in the experiment results. Therefore, we further consider an improved approach that can diminish the above bias. Following negative sampling process in vanilla BPR, we can compute the probability of being negative instance for each item as follows:

$$P(J = j) = \frac{1}{M} \sum_{u \notin \mathcal{S}_j} \frac{1}{N - |\mathcal{S}_u|}. \tag{4}$$

Given the set of $u$ that have interacted with $j$ before, the overall negative sampling probability for $j$ can be directly computed as a sum of $\frac{1}{N-|\mathcal{S}_u|}$, as the sampling process is independent among different users. With the normalization term $\frac{1}{M}$, *i.e.*, an inverse of user count $M$, the sum of $P(J = j)$) over $N$ items equals to 1. Based on this pre-computed item probability distribution, *i.e.*, $P(J)$, the reduced sampling space for users are further generated, which we detail in Algorithm 2. First the item candidates $\mathcal{C}$ with a size of all $\mathcal{R}_u^-$, *i.e.*, $M \times N \times \gamma$, are drawn according to $P(J)$ (Line: 2-6). Then for each user $u$, given $u$'s interacted items $\mathcal{S}_u$, we draw $N \times \gamma$ unobserved instances (without replacement) from $\mathcal{C}$ as the reduced negative sampling space $\mathcal{R}_u^-$ (Line: 9-17). Since the items in above $M$ generated subsets are exactly the same as those in $\mathcal{C}$, which are sampled according to $P(J)$, this approach preserves the distribution characteristics after reducing sampling space and thus diminish the sampling bias existed in the previous approach adopting uniform sampling.

We vary the size ratio $\gamma$ and summarize the performance on all three datasets, Beibei, Tmall-all and Tmall-select, in Table 2. In order to factor out random effects, for each size, we repeat the experiment ten times and save the scores with 100, 80, 60, 40, 20 and 0 iterations left. We finally report the average score, as well as standard variance. For ease of representation, we use term "Uniform" to denote the approach that uniformly generates reduced sampling space, and "Non-uniform" for another approach based on pre-computed distribution. The first row indicates the performance of the original BPR that samples negative items from the whole space.

### 4.2 Results

We first analyze the performance of reduced sampling space that is generated uniformly ("Uniform" ). Surprisingly on the Beibei dataset, except for $\gamma = 1/2^7$, the performance is not decreased but increased after reducing the sampling space. When varying $\gamma$ from $1/2^5$ to $1/2^{10}$, the performance improvement can be at most $1.03\%$ and $0.89\%$ in terms of HR and NDCG, respectively. Even with a rather small $\gamma$ as $1/2^{10}$, where the sampling space of each user only contains $116$ candidates, we still obtain a relative improvement of $0.19\%$ (HR) and $0.44\%$ (NDCG) over the original BPR. This finding is novel and encouraging, meaning that sampling from the whole item space is not only unnecessary for BPR, but may even hurt the performance.

On other two datasets, except for $\gamma = 1/2$ on Tmall-select, we do not observe similar improvements by reducing the sampling space. However, in many cases, both HR and NDCG decrease no less than $1\%$. It needs a fairly small $\gamma$, about $1/2^6$, to cause a significant performance degradation. Also, as we observe some $\Delta$HR or $\Delta$NDCG to be $0.00\%$, this shows that the performance remains equal to that of the original BPR, providing further evidence on the inefficiency of the uniform sampler for BPR.

Then for our second approach that selects the subspace based on the pre-computed distribution $P(J)$ ("Non-uniform"), the result demonstrates its superiority over the uniform sampling one. First of all, on Beibei dataset, all

TABLE 2
Performance of BPR with different settings on the fraction of the reduced sampling space. "Num." means the size of sampling space for each user, *i.e.*, Ratio $\times$ Item#.

| | | Uniform | | | | Non-uniform | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Ratio** | **Num.** | **HR** | **ΔHR** | **NDCG** | **ΔNDCG** | **HR** | **ΔHR** | **NDCG** | **ΔNDCG** |
| $2^0$ | 119012 | $0.1070 \pm 0.0022$ | – | $0.0225 \pm 0.0005$ | – | $0.1070 \pm 0.0022$ | – | $0.0225 \pm 0.0005$ | – |
| $2^{-5}$ | 3719 | $0.1075 \pm 0.0019$ | +0.47% | $0.0226 \pm 0.0004$ | +0.44% | $0.1071 \pm 0.0020$ | +0.09% | $0.0225 \pm 0.0004$ | 0.00% |
| $2^{-6}$ | 1859 | $0.1076 \pm 0.0019$ | +0.56% | $0.0226 \pm 0.0003$ | +0.44% | $0.1077 \pm 0.0018$ | +0.65% | $0.0226 \pm 0.0004$ | +0.44% |
| $2^{-7}$ | 930 | $0.1058 \pm 0.0020$ | −1.12% | $0.0222 \pm 0.0004$ | −1.33% | $0.1080 \pm 0.0019$ | **+0.93%** | $0.0227 \pm 0.0005$ | +0.89% |
| $2^{-8}$ | 465 | $0.1081 \pm 0.0019$ | **+1.03%** | $0.0227 \pm 0.0005$ | **+0.89%** | $0.1078 \pm 0.0012$ | +0.75% | $0.0227 \pm 0.0003$ | +0.89% |
| $2^{-9}$ | 232 | $0.1073 \pm 0.0020$ | +0.28% | $0.0225 \pm 0.0004$ | 0.00% | $0.1070 \pm 0.0025$ | +0.00% | $0.0225 \pm 0.0005$ | 0.00% |
| $2^{-10}$ | 116 | $0.1072 \pm 0.0036$ | +0.19% | $0.0226 \pm 0.0009$ | +0.44% | $0.1080 \pm 0.0022$ | **+0.93%** | $0.0229 \pm 0.0004$ | **+1.78%** |

(a) Beibei

| | | Uniform | | | | Non-uniform | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Ratio** | **Num.** | **HR** | **ΔHR** | **NDCG** | **ΔNDCG** | **HR** | **ΔHR** | **NDCG** | **ΔNDCG** |
| $2^0$ | 32339 | $0.0304 \pm 0.0005$ | – | $0.0076 \pm 0.0002$ | – | $0.0304 \pm 0.0005$ | – | $0.0076 \pm 0.0002$ | – |
| $2^{-2}$ | 8085 | $0.0302 \pm 0.0005$ | −0.66% | $0.0076 \pm 0.0002$ | 0.00% | $0.0304 \pm 0.0006$ | 0.00% | $0.0077 \pm 0.0001$ | **+1.32%** |
| $2^{-3}$ | 4042 | $0.0304 \pm 0.0005$ | 0.00% | $0.0076 \pm 0.0002$ | 0.00% | $0.0302 \pm 0.0004$ | −0.66% | $0.0076 \pm 0.0002$ | 0.00% |
| $2^{-4}$ | 2021 | $0.0302 \pm 0.0005$ | −0.66% | $0.0076 \pm 0.0001$ | 0.00% | $0.0302 \pm 0.0005$ | −0.66% | $0.0077 \pm 0.0002$ | **+1.32%** |
| $2^{-5}$ | 1010 | $0.0302 \pm 0.0004$ | −0.66% | $0.0075 \pm 0.0001$ | −1.32% | $0.0303 \pm 0.0006$ | −0.33% | $0.0076 \pm 0.0002$ | 0.00% |
| $2^{-6}$ | 505 | $0.0300 \pm 0.0005$ | −1.32% | $0.0075 \pm 0.0002$ | −1.32% | $0.0304 \pm 0.0007$ | 0.00% | $0.0076 \pm 0.0002$ | 0.00% |
| $2^{-7}$ | 253 | $0.0300 \pm 0.0004$ | −1.32% | $0.0075 \pm 0.0001$ | −1.32% | $0.0299 \pm 0.0006$ | −1.64% | $0.0075 \pm 0.0002$ | −1.32% |

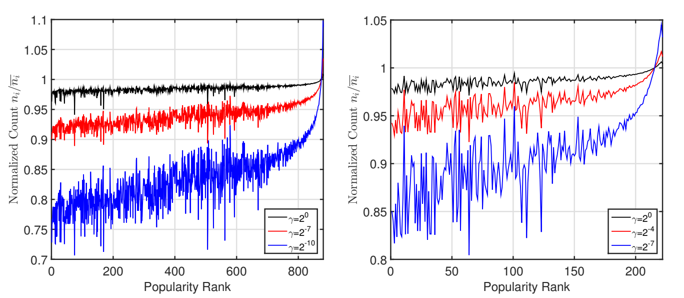
(b) Tmall-all

| | | Uniform | | | | Non-uniform | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Ratio** | **Num.** | **HR** | **ΔHR** | **NDCG** | **ΔNDCG** | **HR** | **ΔHR** | **NDCG** | **ΔNDCG** |
| $2^0$ | 22570 | $0.0744 \pm 0.0015$ | – | $0.0186 \pm 0.0005$ | – | $0.0744 \pm 0.0015$ | – | $0.0186 \pm 0.0005$ | – |
| $2^{-1}$ | 11285 | $0.0746 \pm 0.0015$ | **+0.27%** | $0.0188 \pm 0.0004$ | **+1.08%** | $0.0744 \pm 0.0017$ | 0.00% | $0.0186 \pm 0.0004$ | 0.00% |
| $2^{-2}$ | 5643 | $0.0743 \pm 0.0017$ | −0.13% | $0.0186 \pm 0.0004$ | 0.00% | $0.0750 \pm 0.0017$ | **+0.81%** | $0.0188 \pm 0.0004$ | **+1.08%** |
| $2^{-3}$ | 2821 | $0.0737 \pm 0.0012$ | −0.94% | $0.0186 \pm 0.0003$ | 0.00% | $0.0745 \pm 0.0015$ | +0.13% | $0.0187 \pm 0.0004$ | +0.54% |
| $2^{-4}$ | 1411 | $0.0740 \pm 0.0015$ | −0.54% | $0.0184 \pm 0.0004$ | −1.08% | $0.0741 \pm 0.0013$ | −0.40% | $0.0187 \pm 0.0004$ | +0.54% |
| $2^{-5}$ | 353 | $0.0738 \pm 0.0014$ | −0.81% | $0.0185 \pm 0.0004$ | −0.54% | $0.0738 \pm 0.0014$ | −0.81% | $0.0185 \pm 0.0005$ | −0.54% |
| $2^{-6}$ | 176 | $0.0723 \pm 0.0016$ | −2.82% | $0.0181 \pm 0.0004$ | −2.69% | $0.0737 \pm 0.0017$ | −0.94% | $0.0184 \pm 0.0005$ | −1.08% |

(c) Tmall-select

settings of $\gamma$ bring an equal or improved performance. The maximum increases to $1.78\%$, which is $1.03\%$ in previous approach. Second, for other two Tmall dataset, we observe large improvement on result compared to the previous one. In terms of number of metrics (HR/NDCG) that have decreased after subsampling, both Tmall-all and Tmall-select have 5 metrics, which have decreased from 8 of previous approach. As for the cases where performance is improved after subsampling, the numbers are 2 and 5 on Tmall-all and Tmall-select, respectively, while in previous approach they are 0 and 2, respectively. Moreover, this approach also largely improves the worst performance on several $\gamma$ settings. On Tmall-select, when the size of sampling space is 176, *i.e.*, $\gamma = 1/2^6$, the performance is degraded with about $1\%$ in terms of both HR and NDCG, while those in previous approach are $2.82\%$ and $2.69\%$. Therefore, we can conclude that our approach can diminish the performance degradation under small $\gamma$ and achieve performance improvement on all datasets with a suitable setting of $\gamma$ value. On Beibei dataset, this improved sampler is able to improve the NDCG metric by $1.78\%$ when $\gamma$ is only $1/2^{10}$. As for other two Tmall datasets, the performance improvement is observed when $\gamma$ is above $1/2^4$.

### 4.3 Discussion




(a) Beibei (b) Tmall-all

Fig. 2. The negative sampling count of items with different popularity rank, normalized by the mean value.

As we have illustrated in Table 2, the reduced negative sampling space constantly performs better on Beibei dataset but degrades on Tmall dataset under several settings of reduce ratio $\gamma$, especially when the reduced space is generated by our first approach, *i.e.*, uniform sampling. To investigate the above different observations, we start with analyzing the negative sampling count of items, denoted as $\{n_i\}$, under different settings of $\gamma$, *i.e.*, the number of times that a specific item is sampled as a negative instance during training. Fig. 2 plots the negative sampling count of items during the whole training process on Beibei and Tmall-all, under different $\gamma$. For better illustration we sort the item by their popularity rank and normalize $\{n_i\}$ by their mean value $\overline{n_i}$. On both Fig. 2(a) and (b), we observe that popular items, *i.e.*, with low rank value, are less likely to be sampled as negative instances when $\gamma$ becomes lower. This corresponds to our analysis that uniformly generating subspace would change the negative sampling probability in terms of each item. Since negative sampling is an important step for training BPR, this change on sampling probability can largely impact the performance.

Rendle *et al.* [28] have shown that oversampling popular items as negative feedback underperforms the basic uniform sampler, due to the under-training of those less popular items. Motivated by this, we look back on the popularity skewness of the Beibei and Tmall datasets, which have already been illustrated in Fig. 1. The biggest difference between Beibei and Tmall is that the former is much more skewed, where top-1% of the items accounts for 50% of the purchased interactions, much higher than 10% on Tmall dataset. Therefore, for the original sampler on Beibei, negative instances are sampled from the whole item space ($\sim 10^5$) and large number of unpopular items cannot receive sufficient gradient steps during SGD training. By fixing a reduced sampling space for each user, this ineffectiveness can be diminished with a higher sampling probability for those unpopular items. However, different from Beibei, the popularity skewness of Tmall are much less significant. Though based on whole item space, the vanilla BPR using uniform sampler still trains well on those unpopular items. Therefore, uniformly generating subspace may oversample unpopular items too much and hurt the performance to some extent. Indeed, we also observe that the distribution characteristics of Tmall dataset are more sensitive to this change. More specifically, in Fig. 2, $n_i$ of popular items on Tmall-all ((b), blue curve) are much lower than those on Beibei ((a), red curve), under the same setting of $\gamma = 1/2^7$, indicating these items may be undersampled too much and opposite for those unpopular ones. To summarize, we demonstrate that uniformly generating subspace for negative sampling oversamples the unpopular items and thus works well on a heavily popularity-skewed dataset. This also explains the performance improvement on both Tmall-all and Tmall-select datasets after we adopt another improved approach that manages to maintain the overall distribution characteristics after subsampling.

There still remain several limitations in our proposed reduced sampler. In terms of generating better subspaces for negative sampling, one can consider a more complexed scheme that oversamples unpopular items more adaptively. For example, the previous dynamic negative sampling strategy [28], [40] considers the change of proximity between a user and an item during training, and then selects the most "difficult" negative instance. Motivated by this, a possible solution is to design the dynamic negative subspace that selects both unpopular and under-trained items based on the trained model during the training process. However, this subspace cannot be pre-selected and requires the constant update during the training process, which is highly inefficient and thus beyond the scope of this paper. Comparatively, our proposed random approach is more efficient and achieves performance improvement on all datasets with a suitable setting of reduce ratio value, which can be regarded as a more practicable option. Besides, though we conduct experiments on two real-world datasets, it still requires more extensive experiments on the generality of our observations. However, since there are only a few accessible implicit dataset, we are subject to this limitation and only find Beibei and Tmall dataset with the suitable scale, which are collected from E-commerce websites, a typical implicit recommender system.

To summarize, we have demonstrated that the uniform sampler is unnecessary for BPR and may even degrade the performance, especially in the popularity-skewed datasets. Considering its inefficiency and poor robustness against popularity skewness, we focus on designing a better sampler for BPR in the following sections.

## 5 View-Enhanced Sampler

One inherent issue of recommender systems is the natural scarcity of observed data. To overcome this, BPR samples unobserved items as negative feedback. However, since a user can only interact with a limited number of items, sampling process can be inefficient and may even degrade the performance, as we have demonstrated above. In E-commerce recommender systems, besides the purchase feedback that is directly related to optimizing the conversion rate (CVR), the view logs of users are usually much easier to collect and thus can be leveraged to learn user preference. In this section, we design a view-enhanced sampler for BPR. For readability, we summarize the major notations throughout the paper in Table 3.

### 5.1 Integrating View Signal

Intuitively, viewed interactions can be treated as an intermediate feedback between the purchased and missing interactions. Therefore, for user $u$'s viewed (but not purchased) item $v$, it should have an intermediate value of prediction $\hat{r}_{uv}$ between those of non-viewed item $j$ (*i.e.*, missing entry) and purchased item $i$, *i.e.*, $\hat{r}_{uj}$ and $\hat{r}_{ui}$. Based on this, we propose two variant of BPR sampler that can leverage view data. One is to leverage these viewed items in a biased sampling process, the other is to consider this relationship in a newly proposed objective function.

#### 5.1.1 *Biased Sampling*

First of all, we can integrate the view signal by augmenting the training data. In BPR, a training example $(u, i, j) \in \mathcal{D}$ assumes that $u$ prefers $i$ over $j$. Then, the model parameters, *i.e.*, user vector $\mathbf{p}_u$ and item vector $\mathbf{q}_i$, are updated towards the objective of $\hat{r}_{ui} > \hat{r}_{uj}$. Through a biased sampling process, we are able to encode the intermediate preference information of users' viewed interactions in the model. In our proposed view-enhanced sampler, as illustrated in Fig. 3, we split the item space into three sets for each user $u$, namely $\mathcal{S}_u$, $\mathcal{V}_u$, and $\mathcal{R}_u$, which indicate the purchased items, viewed (but not purchased) items, and remaining

TABLE 3
List of commonly used notations.

| Notation | Description |
|---|---|
| $M, N, K$ | The numbers of users, items, and factors. |
| $\mathbf{P}, \{\mathbf{p}_u\}$ | The latent factor matrix and vector for users. |
| $\mathbf{Q}, \{\mathbf{q}_i\}$ | The latent factor matrix and vector for items. |
| $\mathcal{S}, \mathcal{S}_u$ | The sets of all purchased $(u, i)$ pairs, items purchased by $u$. |
| $\mathcal{V}, \mathcal{V}_u$ | Similar notations for viewed interactions. |
| $\mathcal{R}, \mathcal{R}_u$ | Similar notations for unobserved interactions. |
| $\hat{r}_{ui}, \hat{r}_{uv}, \hat{r}_{uj}$ | Predictions of user $u$ over purchased items $i$, viewed items $v$ and non-viewed items $j$. |
| $\{\omega_1, \omega_2, \omega_3\}$ | Probability of sampling training item pairs. |
| $\alpha$ | Weight of training pairs made up of a purchased item and a viewed item. |
| $\alpha_u$ | User-oriented weight of training pairs made up of a purchased item and a viewed item. |
| $\beta$ | Significance level of view-purchase ratio in $\alpha_u$. |
| $\lambda$ | Regularization parameter. |

non-viewed items, respectively. Then, we sample an item pair from three candidate sets, $\{(i, v)|i \in \mathcal{S}_u, v \in \mathcal{V}_u\}$, $\{(i, j)|i \in \mathcal{S}_u, j \in \mathcal{R}_u\}$, and $\{(v, j)|v \in \mathcal{V}_u, j \in \mathcal{R}_u\}$, with predefined probabilities $[\omega_1, \omega_2, \omega_3]$ respectively, where $\omega_1 + \omega_2 + \omega_3 = 1$. The generated training example is finally used to update the model parameters in (1) (see Ref. [29] for further details). We term the BPR method with this view-enhanced sampler as *BPR+view*$_{prob}$.

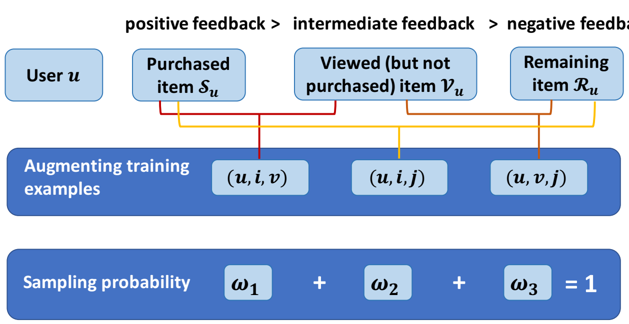



Fig. 3. Biased sampling process considering users' viewed items.

Our proposed *BPR+view*$_{prob}$ uses biased sampling to exploit the side information provided by the viewed items. As each training example in *BPR+view*$_{prob}$ only contains two items, the viewed items are sampled as negative feedback and positive feedback with a probability of $\omega_1$ and $\omega_3$, respectively. In other words, it is hard to jointly learn the two-fold semantics of user preference on these viewed items, which assigns them a positive signal compared to those non-viewed items and a negative signal compared to purchased ones. Therefore, next we move forward to improve BPR sampler by considering a view-enhanced weighted loss in objective function.

#### 5.1.2 Weighted Loss

To overcome the inefficacy issue in *BPR+view*$_{prob}$, we propose to sample an item triple $(i, v, j)$ in each training example, where $i$, $v$ and $j$ represent a user's purchased item, viewed item and non-viewed item, respectively. Considering the user preference on these three items, the model parameters, $\{\mathbf{p}_u, \mathbf{q}_i, \mathbf{q}_v, \mathbf{q}_j\}$, should be updated towards the objective of $\hat{r}_{ui} > \hat{r}_{uv} > \hat{r}_{uj}$. Therefore, similar to BPR, we design following objective function:

$$J(\Theta) = \arg\min_{\Theta} \sum_{(u,i,v,j)\in\mathcal{D}} - \ln \sigma(\hat{r}_{ui}(\Theta) - \hat{r}_{uj}(\Theta)) \\ - \alpha \ln \sigma(\hat{r}_{ui}(\Theta) - \hat{r}_{uv}(\Theta)) - (1 - \alpha) \ln \sigma(\hat{r}_{uv}(\Theta) - \hat{r}_{uj}(\Theta)), \tag{5}$$

where $\sigma(x) = 1 - \sigma(x)$, and $\Theta$ denotes the set of all parameters to be optimized. All three pairwise ranking relations among $i$, $v$ and $j$ are considered. Since the viewed item $v$ can be considered as both negative ($\hat{r}_{ui} > \hat{r}_{uv}$) and positive ($\hat{r}_{uv} > \hat{r}_{uj}$) feedback, the weighting parameter $\alpha$ in (5) controls the relative strength between these two semantics. Therefore, by tuning $\alpha$ empirically, we can train a model that properly exploits the user preference of view signal. Compared with *BPR+view*$_{prob}$, this sampler simultaneously draws a purchased item $i$, a viewed item $v$ and an unobserved item $j$ for each user $u$. It is noteworthy that when $\alpha = 0$ or $\alpha = 1$, users' viewed interactions are considered as positive or negative signal only, following the similar assumption to the case of $[\omega_1, \omega_2, \omega_3] = [0, 0.5, 0.5]$ or $[\omega_1, \omega_2, \omega_3] = [0.5, 0.5, 0]$ in *BPR+view*$_{prob}$. However, these two samplers differ in that *BPR+view*$_{prob}$ handles $(u, i, j, v)$ in two independent samples, while the other jointly trains $(u, i, j, v)$ in one sample. We further show their difference in terms of the performance results in the experiment.

Note that we have omitted $L_2$ regularization terms for clarity. We use matrix factorization to predict $\hat{r}_{ui}$, user $u$'s preference on item $i$, obtained by calculating the dot product of the latent factors of the user $\mathbf{p}_u$ and the item $\mathbf{q}_i$, as follows:

$$\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i = \sum_{f=1}^{K} p_{u,f} \times q_{i,f}. \tag{6}$$

Recall that $K$ is the number of latent factors. Finally, we use Stochastic Gradient Descent (SGD) to find a local

**Algorithm 3:** Learning Algorithm for *BPR+view*$_{loss}$.

```
Input  : purchase data S, view data V
Output: Θ = {P ∈ R^{M×K}, Q ∈ R^{N×K}}
Randomly initialize P and Q;
while not reaching convergence do
    // Random sampling
    u ← draw a random user from U
    i ← draw a random purchased item from S_u
    v ← draw a random viewed item from V_u
    j ← draw a random non-viewed item from R_u
    // Eq. (8) - (11)
    Compute gradients of {p_u, q_i, q_v, q_j}
    // Eq. (7)
    Update the above parameters
end
```

minimum of the objective function in (5). In particular, for each iteration (Algorithm 3, Lines: 3-11), given a random feedback triple of user $u$ who has purchased item $i$, viewed (but not purchased) item $v$ but not viewed item $j$, $(u,i,v,j) \in \mathcal{D} = \{(u,i,v,j) | i \in \mathcal{S}_u \wedge v \in \mathcal{V}_u \wedge j \in \mathcal{R}_u\}$, we update the model parameter $\theta \in \Theta$ based on the gradient of its corresponding parameter $\frac{\partial J}{\partial \theta}$ while fixing the others, until convergence, as follows:

$$\theta^{(t+1)} = \theta^{(t)} + \eta^{(t)} \cdot \frac{\partial J}{\partial \theta}(\theta^{(t)}). \tag{7}$$

Note that learning rate parameter $\eta$ can both be a fixed constant or an adaptive value like Adagrad [9]. The gradients of latent vectors $\{\mathbf{p}_u, \mathbf{q}_i, \mathbf{q}_v, \mathbf{q}_j\}$ are calculated as follows:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{p}_u} &= \delta(\hat{r}_{ui} - \hat{r}_{uj})(\mathbf{q}_i - \mathbf{q}_j) + \alpha\delta(\hat{r}_{ui} - \hat{r}_{uv})(\mathbf{q}_i - \mathbf{q}_v) \\ &\quad + (1-\alpha)\delta(\hat{r}_{uv} - \hat{r}_{uj})(\mathbf{q}_v - \mathbf{q}_j) - \lambda \mathbf{p}_u,\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{q}_i} &= \delta(\hat{r}_{ui} - \hat{r}_{uj})\mathbf{p}_u + \alpha\delta(\hat{r}_{ui} - \hat{r}_{uv})\mathbf{p}_u \\ &\quad + (1-\alpha)\delta(\hat{r}_{uv} - \hat{r}_{uj})\mathbf{p}_u - \lambda \mathbf{q}_i,\end{aligned} \tag{9}$$

$$\frac{\partial J}{\partial \mathbf{q}_v} = -\alpha\delta(\hat{r}_{ui} - \hat{r}_{uv})\mathbf{p}_u + (1-\alpha)\delta(\hat{r}_{uv} - \hat{r}_{uj})\mathbf{p}_u - \lambda \mathbf{q}_v, \tag{10}$$

$$\frac{\partial J}{\partial \mathbf{q}_j} = -\delta(\hat{r}_{ui} - \hat{r}_{uj})\mathbf{p}_u - (1-\alpha)\delta(\hat{r}_{uv} - \hat{r}_{uj})\mathbf{p}_u - \lambda \mathbf{q}_j, \tag{11}$$

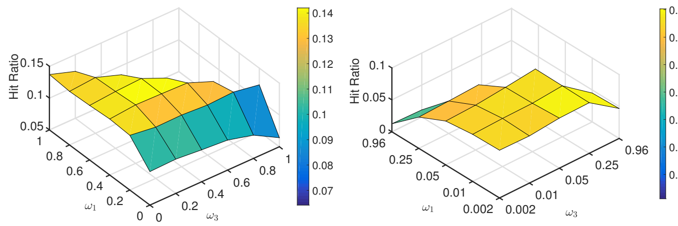
where the regularization parameter $\lambda$ is added to avoid overfitting. Regarding the complexity of the above pairwise learning algorithm, the computation of each gradient is $O(K)$, where $K$ is the number of latent factors. The total complexity in each iteration is $O(T \cdot K)$, where $T$ is the number of training examples. We term the above variant of BPR sampler as *BPR+view$_{loss}$*.

### 5.2 User-aware Weighting Strategy

In *BPR+view$_{loss}$*, a viewed interaction is simultaneously considered as negative feedback compared with the purchased interaction and positive one compared with the unobserved interaction, tuning by a hyperparameter $\alpha$. Intuitively, if a user tend to view many items and instead purchase another one, the viewed interactions should indicate a stronger negative signal than that of other users. In this sense, the relative strength between two semantics of view signal, *i.e.*, $\alpha$, should differs among users. Let $A_u$ denote a user $u$'s view-purchase ratio that measures the degree of whether $u$ prefers to view many items before deciding which to buy. It is reasonable to think that a user with high $A_u$ only has a low interest on those viewed items, which corresponds to a higher weight $\alpha$ in our proposed *BPR+view$_{loss}$*. To account for this effect, we parametrize a user-oriented weight $\alpha_u$ based on $A_u$:

$$\alpha_u = \frac{A_u^\beta}{(A_u^\beta + 1)}, \tag{12}$$

where the high value of view-purchase ratio $A_u$ would get a high $\alpha$ close to 1, and the exponent $\beta$ controls the significance level of this effect. This design of $\beta$ is inspired by




(a) Beibei (b) Tmall-select

Fig. 4. Impact of sampling probability parameters $\{\omega_1, \omega_2, \omega_3\}$ on *BPR+view$_{prob}$*'s performance, in terms of HR.

previous works that consider to smoothen the popularity-based weight in negative sampling [12], [23]. We term this new BPR sampler with user-aware weighting scheme as *BPR+view$^{\beta}_{loss}$*

Next, we focus on the definition of view-purchase ratio $A_u$ above. A straightforward way of computing it would be the ratio between number of user $u$'s viewed interactions and purchased ones. However, as users' shopping history is divided into several sessions, computing $A_u$ in the session-level can be more accurate. More specifically, we define $A_u$ as the average value among these sessions:

$$A_u = \frac{\sum_{s=1}^{S} a_{u,s}}{|S|}, \; a_{u,s} = \frac{\mathcal{V}_{u,s}}{\mathcal{P}_{u,s}}, \tag{13}$$

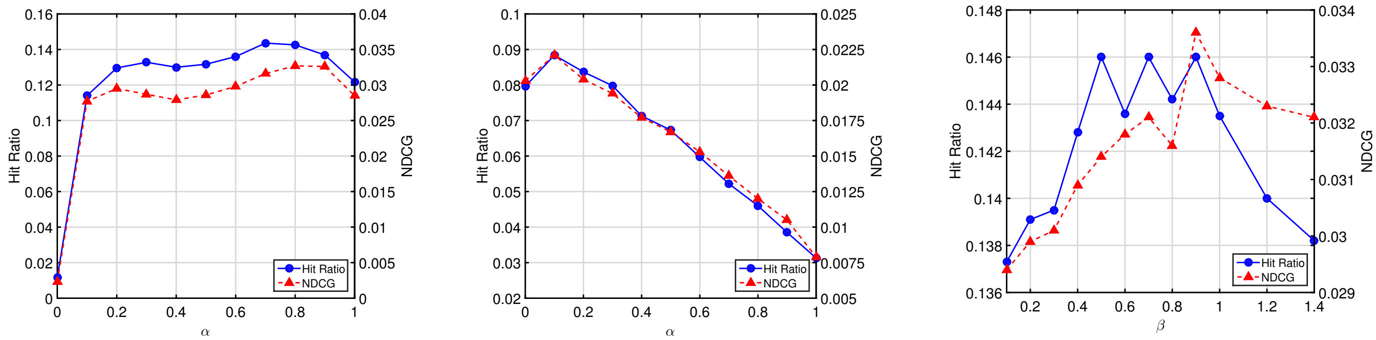
where $a_{u,s}$, $\mathcal{V}_{u,s}$ and $\mathcal{R}_{u,s}$ represent $u$'s view-purchase ratio, viewed item set and purchased item set in session $s$, respectively. To generate $u$'s sessions in the shopping history, we first sort $u$'s viewed and purchased interactions according to timestamps and then we merge those consecutive interactions into one session based on whether they happen within a threshold $d$. Since the suitable setting of $d$ may vary between different datasets, we empirically tune this parameter and search the best recommendation performance. The result shows that $d = 3600$ (s) works well in Beibei dataset. As for Tmall dataset, since the timestamp information only contains the date, it is infeasible to extract session information in each user's shopping history. Therefore, we leave the exploration of user-aware weighting scheme on *BPR+view$^{\beta}_{loss}$* for future work.

### 5.3 Results

We first study the influence of hyper-parameters. Then we analyze the performance gain of our view-enhanced BPR sampler. Finally we compare with the state-of-the-art baselines.

#### 5.3.1 Hyper-parameter Investigation

***BPR+view$_{prob}$*.** In the biased sampling, our proposed *BPR+view$_{prob}$* has three non-negative parameters: $[\omega_1, \omega_2, \omega_3]$, which respectively represents the probability of item pairs among users' purchased, viewed and unobserved interactions. Considering $\omega_1 + \omega_2 + \omega_3 = 1$, we have to search two independent parameters. Fig. 4 shows its performance (HR) with different settings of $\{\omega_1, \omega_3\}$. In Fig. 4(a), we visualize the results on Beibei with a grid search in $[0, 0.2, 0.4, 0.6, 0.8, 1.0]$, and the relatively higher HR is




(a) *BPR+view$_{loss}$*, Beibei (b) *BPR+view$_{loss}$*, Tmall-select (c) *BPR+view$^{\beta}_{loss}$*, Beibei

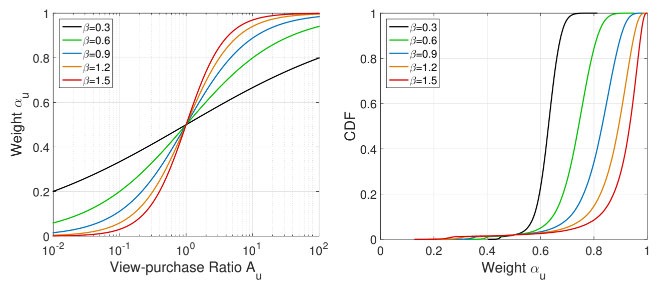
Fig. 5. Impact of weighting parameters $\alpha$ and $\beta$ on HR performances of *BPR+view$_{loss}$* and *BPR+view$^{\beta}_{loss}$*, respectively.

observed between 0.2 and 0.6. Our further fine-grained tuning locates the best setting at $[\omega_1, \omega_2, \omega_3] = [0.3, 0.3, 0.4]$. In terms of the two-fold semantics encoded in view data, we use $\frac{\omega_3}{\omega_1}$ to measure whether it is more closed to positive feedback or negative feedback. Here in Beibei, $\frac{\omega_3}{\omega_1}$ is close to 1, indicating both two folds are important. We further investigate $[\omega_1, \omega_2, \omega_3]$ in Tmall-select and observe that peak performance lies in $[0.01, 0.74, 0.25]$, as shown in Fig. 4(b). In a word, the similar effect of view data is observed between Beibei and Tmall-select, while the only difference lies in that Tmall users' viewing behavior is much closer to a positive feedback, with a larger value of $\frac{\omega_3}{\omega_1}$.

***BPR+view$_{loss}$*.** Now, we study the impact of weighting parameter $\alpha$ on *BPR+view$_{loss}$*. As shown in Fig. 5(a), we observe the best $\alpha$ varies between 0.7 and 0.8 on Beibei. Since a large $\alpha$ increases the importance of learning user preference from purchased and viewed item pairs, this observation highlights the significance of considering users' viewing behavoirs more as a negative feedback. However, the performance still shows a drop at $\alpha = 1$, where we take viewed items as equally important as those unobserved ones. This observation also confirms the necessity of taking viewed interactions as a weak positive feedback. In Fig. 5(b), the performance drop steeply as $\alpha$ increases in Tmall-select dataset and the peak lies at $\alpha = 0.1$, where viewed items are almost utilized equally as purchased ones but pairwise ranking relation between them still exists. The performance of *BPR+view$_{loss}$* is sensitive to $\alpha$ in Tmall-select, while not in Beibei. This difference may be caused by the same reason as distinctive influence of $[\omega_1, \omega_2, \omega_3]$ on *BPR+view$_{prob}$* mentioned above, that view data represents a more effective signal of user preference in Tmall-select dataset.

***BPR+view$^{\beta}_{loss}$*.** Fig. 5(c) plots the prediction accuracy of *BPR+view$^{\beta}_{loss}$* on Beibei, with different $\beta$. Though HR increases to its maximum at 3 different $\beta$, this model achieves best performance at $\beta = 0.9$ evaluated by both HR and NDCG[7]. To further explain the advantage of user-oriented weight $\alpha_u$ over a uniform weight $\alpha$ as used in *BPR+view$_{loss}$*, we first plot the $\alpha_u$ curves under different settings of $\beta$ in Fig. 6(a), which illustrates the strengthening effect of $\beta$ on assigning view-rather-than-buy users (*i.e.*, with a high

7. Note that we have saved scores with 100, 80, 60, 40, 20 and 0 iterations left, then reported the mean values.




(a) (b)

Fig. 6. (a) User-oriented weight values versus view-purchase ratio under different settings of significance exponent. (b) Distribution of user-oriented weight under different settings of significance exponent.

$A_u$) a large weight $\alpha_u$. Then, in Fig. 6(b), we plot the CDF of $\alpha_u$ under different $\beta$. Under the best setting as $\beta = 0.9$, the mean value and median value of $\alpha_u$ is 0.81 and 0.83, respectively, which is close to our observation in Fig. 5(a) that best $\alpha$ is between 0.7 and 0.8. With ability to vary among different users, *BPR+view$^{\beta}_{loss}$* sampler with $\alpha_u$ outperforms that with the uniform $\alpha$. As for Tmall-select, since we cannot extract users' shopping sessions from the coarse-grained timestamp in each record, we do not conduct similar experiments on this dataset.

According to the investigation above, we fix these hyper-parameters according to the best performance evaluated by HR, *i.e.*, $[\omega_1, \omega_2, \omega_3] = [0.3, 0.3, 0.4], \alpha = 0.7, \beta = 0.9$ for Beibei and $[\omega_1, \omega_2, \omega_3] = [0.01, 0.74, 0.25], \alpha = 0.1$ for Tmall-select.

#### 5.3.2 *Performance Gain of View-Enhanced Sampler*

We compare the performance of vanilla *BPR* and our proposed view-enhanced sampler. The main result is listed in Table 4.

***BPR+view$_{prob}$* vs. *BPR*.** To demonstrate the effectiveness of our proposed *BPR+view$_{prob}$*, we compare it with 1) the vanilla *BPR* [29], and 2) *BPR-DNS* [40], which selects the item with the highest prediction score among $X$ randomly sampled negatives. For *BPR-DNS*, we tune the $X$ in the same way as the original paper. To our knowledge, DNS is the most effective sampler to date for BPR based on the interaction data only, and empirically outperforms [28]. In

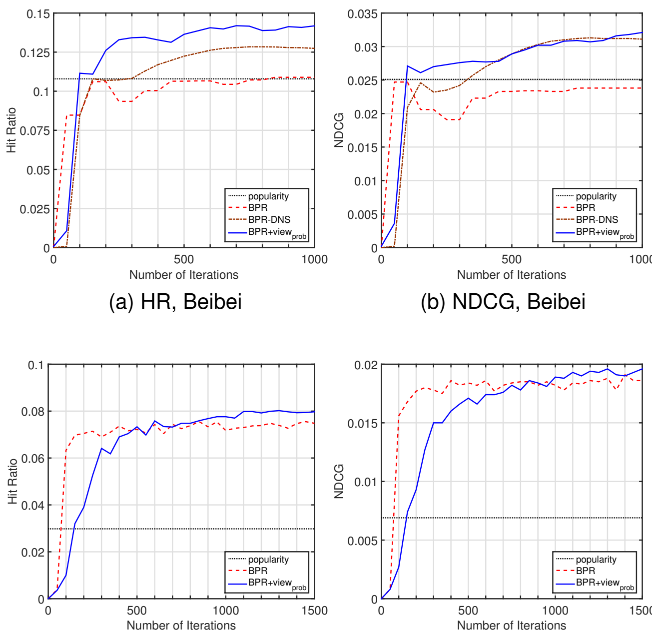
TABLE 4
Performance gain of our proposed sampler, *i.e.*, *BPR+view$_{prob}$*, *BPR+view$_{loss}$* and *BPR+view$^{\beta}_{loss}$*.

| | HR | Δ | NDCG | Δ |
|---|---|---|---|---|
| **BPR (baseline)** | 0.1086 | – | 0.0242 | – |
| **BPR+view$_{prob}$** | 0.1422 | +30.93% | 0.0321 | +32.64% |
| **BPR+view$_{loss}$** | 0.1436 | +32.23% | 0.0327 | +35.12% |
| **BPR+view$^{\beta}_{loss}$** | 0.1460 | +34.44% | 0.0336 | +38.84% |

(a) Beibei

| | HR | Δ | NDCG | Δ |
|---|---|---|---|---|
| **BPR (baseline)** | 0.0755 | – | 0.0191 | – |
| **BPR+view$_{prob}$** | 0.0807 | +6.89% | 0.0199 | +4.19% |
| **BPR+view$_{loss}$** | 0.0884 | +17.09% | 0.0221 | +15.71% |

(b) Tmall-select



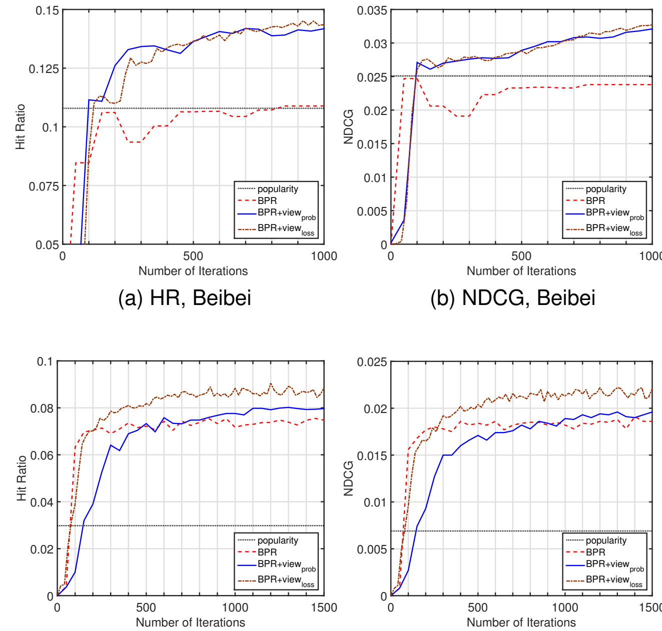

(a) HR, Beibei (b) NDCG, Beibei
(c) HR, Tmall-select (d) NDCG, Tmall-select

Fig. 7. Performance comparison in each iteration (*BPR+view$_{prob}$*).

addition, we also evaluate a common baseline *Popularity*, which simply recommends items based on their popularity evidenced by the number of purchases.

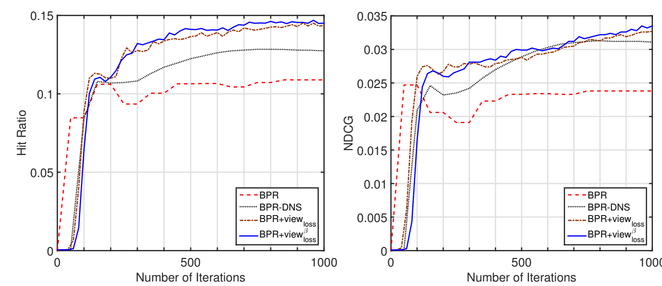
Fig. 7 shows the testing HR and NDCG of the compared methods in each training iteration. As can be seen, upon convergence, *BPR+view$_{prob}$* significantly outperforms all other methods on three datasets, except for the NDCG on Beibei. Its NDCG is 0.0321, while *BPR-DNS* gets 0.0313, about 2.50% better. This justifies the efficacy of accounting for the preference signal in the view data using our proposed sampler. Besides, the relative improvements of *BPR+view$_{prob}$* over *BPR* are about 30%+ and 5%+ on Beibei and Tmall-select dataset, respectively (See Table 4). Last but not least, we observed that *Popularity* performs as well as *BPR* on the Beibei dataset, which is unexpected since *BPR* is a personalized recommendation method. Our further investigation finds that it is because the Beibei dataset is highly popularity-skewed — the top-1% items contribute almost 50% of purchases, as illustrated in Fig. 1(a).

Clearly, after integrating view signal as intermediate feedback, *BPR+view$_{prob}$* outperforms the original *BPR* that only contains purchase feedback.




(a) HR, Beibei (b) NDCG, Beibei
(c) HR, Tmall-select (d) NDCG, Tmall-select

Fig. 8. Performance comparison in each iteration (*BPR+view$_{loss}$*).




(a) HR, Beibei (b) NDCG, Beibei

Fig. 9. Performance comparison in each iteration (*BPR+view$^{\beta}_{loss}$*).

***BPR+view$_{loss}$* vs. *BPR+view$_{prob}$*.** To evaluate our two proposed variants of BPR sampler, *i.e.*, biased sampling scheme and weighted loss scheme, we look further into the comparison of *BPR+view$_{prob}$* and *BPR+view$_{loss}$* for every iteration, in Fig. 8. For Beibei, the relative improvement in terms of HR and NDCG are 1.29% and 2.48% respectively (0.1436 vs. 0.1422 and 0.0327 vs. 0.0321, Table 5). Moreover, for Tmall-select, we observe a relative improvement of 10.20% (0.0884 vs. 0.0807) and 11.52% (0.0221 vs. 0.0199) on two evaluation indexes, which indicates the stronger influence of viewing behavoirs on Tmall again. The obvious improvements demonstrates that considering three pairwise relations among the sampled item triple (a purchased item, a viewed item and an unobserved item) can better describe both positive and negative signals of viewing behaviors. Even though *BPR+view$_{prob}$* outperforms vanilla *BPR* and *BPR-DNS*, it still has difficulty in treating viewed interactions as both positive and negative feedback in a single sampling.

***BPR+view$_{loss}$* vs. *BPR+view$^{\beta}_{prob}$*.** Finally, we compare the performance of $BPR$+$view_{loss}$ and $BPR$+$view^{\beta}_{loss}$ in Fig. 9 to evaluate the efficacy of user-oriented weighting scheme. On Beibei dataset, by imposing personalized weighting strategy, $BPR$+$view^{\beta}_{loss}$ achieves a further relative improvement of 1.67% (0.1460 vs. 0.1436) and 2.75% (0.0336 vs. 0.0327) w.r.t. HR and NDCG, which proves our intuition that viewed interactions indicate stronger negative signal for users with larger view-purchase ratio.

To summarize, modelled as an intermediate feedback, users' viewed interactions can play an important role in learning a more precise user preference to improve recommendation performance. Compared with integrating view signal through a biased sampler, simultaneously learning two-fold semantics of view signal in each update step performs much better. By taking into account the effect of users' online-shopping habits, we design a user-oriented weighting scheme which achieves further improvements.

### 5.3.3 Performance Comparison

**Baselines**. Besides vanilla BPR, we also consider following baseline methods:

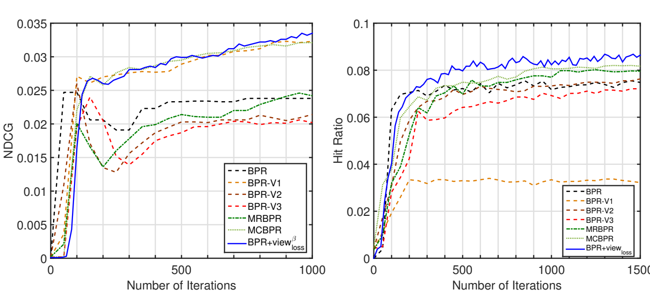
**- NeuMF** [11]. Neural Matrix Factorization is a state-of-the-art neural-network based method for implicit recommender systems. It combines MF and multi-layer perceptrons (MLP) to learn the user-item interaction function. As suggested in the paper, we adopted BPR loss, pre-trained the model with MF, and tuned the depth and $L_2$ regularizer for the hidden layers.

**- BPR Variant**. We also implemented three BPR variants that use users' view logs in negative sampler. The first variant (BPR-V1) considers users' viewed items and unobserved items as equal, *i.e.*, each with half the chance in negative sampling. The second one (BPR-V2) ignores all the views and only samples negative instances from unobserved items. The last one (BPR-V3) uses the viewed items in a same way as purchased items, both as the positive instances during training.

**- MR-BPR** [16]. Applying CMF technique to BPR, this method is able to exert the impact of viewing behavior on predicting purchases.

**- MC-BPR** [21]. This method leverages level information of view logs when sampling negative items.

This set of baselines is all based on BPR approach and stands for the state-of-the-art performance. In particular, NeuMF is the recently proposed neural recommender model which has shown significant improvements over conventional shallow methods. As for the methods that integrate both purchase and view data, we choose MR-BPR and MC-BPR, as well as three variants of vanilla BPR. For the above baselines, we have carefully explored the corresponding parameters. As for the learning rate and regularization $\lambda$, all baselines are tuned similarly as mentioned in Sec. 3.4. Except for NeuMF that starts training from a pre-trained model, we run all other methods for 1000 iterations and 1500 iterations on Beibei and Tmall dataset, respectively, which are enough for them to converge. To better illustrate the training process, we also plot the prediction accuracy of each method in each training iteration in Fig. 10.




(a) NDCG, Beibei (b) HR, Tmall-select

Fig. 10. Performance comparison in each iteration (baseline).

TABLE 5
Performance comparison with baseline methods.

| Behavior | Method | HR | Δ | NDCG | Δ |
|---|---|---|---|---|---|
| **Purchase** | **BPR** | 0.1086 | +34.44% | 0.0242 | +38.84% |
| | **NeuMF** | 0.1158 | +26.08% | 0.0277 | +21.30% |
| **Purchase View** | **BPR-V1** | 0.1384 | +5.49% | 0.0320 | +5.00s% |
| | **BPR-V2** | 0.1029 | +41.89% | 0.0214 | +57.01% |
| | **BPR-V3** | 0.0988 | +47.77% | 0.0207 | +62.32% |
| | **MR-BPR** | 0.1119 | +30.47% | 0.0244 | +37.70% |
| | **MC-BPR** | 0.1430 | +2.10% | 0.0320 | +5.00% |
| | $BPR$+$view^{\beta}_{loss}$ | **0.1460** | – | **0.0336** | – |

(a) Beibei

| Behavior | Method | HR | Δ | NDCG | Δ |
|---|---|---|---|---|---|
| **Purchase** | **BPR** | 0.0755 | +17.09% | 0.0191 | +15.71% |
| | **NeuMF** | 0.0785 | +12.61% | 0.0192 | +15.10% |
| **Purchase View** | **BPR-V1** | 0.0322 | +174.53% | 0.0087 | +154.02% |
| | **BPR-V2** | 0.0763 | +15.86% | 0.0190 | +16.32% |
| | **BPR-V3** | 0.0721 | +22.61% | 0.0178 | +24.16% |
| | **MR-BPR** | 0.0796 | +11.06% | 0.0193 | +14.51% |
| | **MC-BPR** | 0.0817 | +8.20% | 0.0201 | +9.95% |
| | $BPR$+$view_{loss}$ | **0.0884** | – | **0.0221** | – |

(b) Tmall-select

The main results are listed in Table 5. On both Beibei and Tmall-select datasets, our proposed sampler outperforms the state-of-the-art baseline methods by a large margin. More specifically, in terms of HR, the relative improvements are 2.10% and 8.20%, *i.e.*, 0.1460 v.s. 0.1430 and 0.0884 v.s. 0.0817, respectively. And for NDCG, we observe an improvement of 5.00% and 9.95%, respectively. With a relative improvement of 2.10∼9.95% *w.r.t* HR and NDCG on two datasets, our proposed sampler can learn more accurate user preference from viewing behaviors. On the one hand, NeuMF significantly improves recommendation accuracy compared with vanilla BPR, yet still worse than ours. This indicates the importance of including additional information inside the view logs, which cannot be complemented by the sophisticated design of a deep model. On the other hand, compared with MC-BPR and MR-BPR that also consider difference between viewed items and unobserved items, our proposed view-enhanced sampler is much more effective because it simultaneously models pairwise ranking relations among user's purchased, viewed and unobserved items in each training example.

Furthermore, it is noteworthy that three BPR variants perform differently between two datasets. Corresponding to our previous analysis of $\omega$ parameters in *BPR+view*$_{prob}$, this is due to the fact that users' viewed interactions indicate stronger negative preference signal on Beibei and by contrast indicate stronger positive preference signal on Tmall-select. For example, BPR-V1 works significantly better than BPR on Beibei but works the opposite on Tmall-select. As it considers users' viewed items and unobserved items as equal in negative sampling, performance on Tmall-select degrades significantly. As for BPR-V2, removing the view data from negative signal slightly increases performance on Tmall-select. However, directly considering view data as positive signal, *i.e.*, BPR-V3, cannot improve performance even on Tmall-select, implying the necessity to consider two-fold semantics in view data. Besides, we observe that the performance of BPR-V3 and BPR-V1 are different with that of *BPR+view*$_{loss}$ when $\alpha = 0$ and $\alpha = 1$ (Fig. 5), respectively, which corresponds to our previous analysis of the difference between *BPR+view*$_{loss}$ and *BPR+view*$_{prob}$.

### 5.4 Discussion

Motivated by the assumption that users' viewing behaviors in E-commerce websites have two-fold semantics, we design the view-enhanced BPR sampler that can better model user preference among the purchased, viewed and unobserved items. Through extensive experiments on two real-world datasets, we observe the performance improvement on not only our proposed sampler, but also other baseline methods that use view data. This demonstrates the advantage of incorporating users' viewing behavior into BPR framework, which is guaranteed by the fact that these view logs do have additional information about user preference. In this sense, our proposed sampler is a better design of learning the inherent nature of user preference. As for the generality of view-enhanced sampler, on the one hand, users' view actions are general and highly frequent in today's online information systems where users interact with commodities, ads, scientific articles and so on. Therefore, it is important to learn more accurate user preference by integrating view data. On the other hand, the idea of modeling ranking relations among different feedbacks in our proposed sampler is general, making it adaptable for other user feedbacks.

However, our experiments are subject to some limitations such as the scale of the data, the off-line evaluation and so on, which may impact the generality of our conclusions to some extent. Thus, the real-world scenario testing is still required. Moreover, although our proposed sampler is adaptable for other intermediate user feedbacks similar to view, the performance gain still requires further investigation.

## 6 Conclusion and Future Work

This paper studied the problem of improving BPR sampler in implicit feedback recommender systems. First, we have demonstrated that sampling negative items from the whole space is unnecessary for BPR. Then, to further improve BPR sampler's ability of learning user preference, we propose an enhanced sampler that encodes two-fold semantics in user's viewing behaviors. With these design, our improved BPR sampler is able to achieve higher accuracy.

This work has focused on collaborative filtering setting, which only leverages the feedback data and is mostly used in the candidate selection stage of industrial recommender systems [34], [36]. In future, we will focus more on the ranking stage, integrating view data into generic feature-based models, such as expressive neural factorization machines [10] and more explainable tree-enhanced embedding model [35].

## References


[1] U. Akram, P. Hui, M. K. Khan, M. Hashim, Y. Qiu, and Y. Zhang. Online impulse buying on double eleven shopping festival: An empirical investigation of utilitarian and hedonic motivations. In *ICMSEM*, pages 680–692. Springer, 2017.

[2] I. Bayer, X. He, B. Kanagal, and S. Rendle. A generic coordinate descent framework for learning from implicit feedback. In *WWW*, pages 1341–1350, 2017.

[3] C. Cao, H. Ge, H. Lu, X. Hu, and J. Caverlee. What are you known for?: Learning user topical profiles with implicit and explicit footprints. In *SIGIR*, pages 743–752, 2017.

[4] J. Cheng, T. Yuan, J. Wang, and H. Lu. Group latent factor model for recommendation with multiple user behaviors. In *SIGIR*, pages 995–998, 2014.

[5] C. J. Cobb and W. D. Hoyer. Planned versus impulse purchase behavior. *Journal of retailing*, 1986.

[6] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *RecSys*, pages 191–198, 2016.

[7] C. Cui, J. Shen, L. Nie, R. Hong, and J. Ma. Augmented collaborative filtering for sparseness reduction in personalized poi recommendation. *ACM Transactions on Intelligent Systems and Technology*, 8(5):71, 2017.

[8] A. F. Da Costa and M. G. Manzato. Exploiting multimodal interactions in recommender systems with ensemble algorithms. *Information Systems*, 56:120–132, 2016.

[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[10] X. He and T.-S. Chua. Neural factorization machines for sparse predictive analytics. In *SIGIR*, pages 355–364, 2017.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.

[12] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*, pages 549–558, 2016.

[13] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[14] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

[15] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[16] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *WSDM*, pages 173–182, 2012.

[17] L. Lerche and D. Jannach. Using graded implicit feedback for bayesian personalized ranking. In *RecSys*, pages 353–356, 2014.

[18] J. Liu, C. Shi, B. Hu, S. Liu, and S. Y. Philip. Personalized ranking recommendation via integrating multiple feedbacks. In *PAKDD*, pages 131–143, 2017.

[19] Y. Liu, H. Li, G. Peng, B. Lv, and C. Zhang. Online purchaser segmentation and promotion strategy selection: evidence from chinese e-commerce market. *Annals of Operations Research*, 233(1):263–279, 2015.

[20] C. Lo, D. Frankowski, and J. Leskovec. Understanding behaviors that lead to purchasing: A case study of pinterest. In *KDD*, pages 531–540. ACM, 2016.

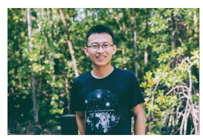
[21] B. Loni, R. Pagano, M. Larson, and A. Hanjalic. Bayesian personalized ranking with multi-channel user feedback. In *RecSys*, pages 361–364, 2016.

[22] L. Mei, P. Ren, Z. Chen, L. Nie, J. Ma, and J.-Y. Nie. An attentive interaction network for context-aware recommendations. In *CIKM*, pages 157–166, 2018.
[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
[24] L. Nie, X. Wei, D. Zhang, X. Wang, Z. Gao, and Y. Yang. Data-driven answer selection in community qa systems. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1186–1198, 2017.
[25] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, pages 502–511, 2008.
[26] W. Pan, H. Zhong, C. Xu, and Z. Ming. Adaptive bayesian personalized ranking for heterogeneous implicit feedbacks. *Knowledge-Based Systems*, 73:173–180, 2015.
[27] H. Qiu, Y. Liu, G. Guo, Z. Sun, J. Zhang, and H. T. Nguyen. Bprh: Bayesian personalized ranking for heterogeneous implicit feedback. *Information Sciences*, 453:80–98, 2018.
[28] S. Rendle and C. Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*, pages 273–282, 2014.
[29] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.
[30] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *KDD*, pages 650–658, 2008.
[31] B. Song, X. Yang, Y. Cao, and C. Xu. Neural collaborative ranking. In *CIKM*, pages 1353–1362, 2018.
[32] X. Song, F. Feng, J. Liu, Z. Li, L. Nie, and J. Ma. Neurostylist: Neural compatibility modeling for clothing matching. In *MM*, pages 753–761, 2017.
[33] L. Tang, B. Long, B.-C. Chen, and D. Agarwal. An empirical study on recommendation with multiple types of feedback. In *KDD*, pages 283–292, 2016.
[34] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD*, pages 839–848, 2018.
[35] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua. Tem: Tree-enhanced embedding model for explainable recommendation. In *WWW*, pages 1543–1552, 2018.
[36] Z. Wang, Z. Jiang, Z. Ren, J. Tang, and D. Yin. A path-constrained framework for discriminating substitutable and complementary products in e-commerce. In *WSDM*, pages 619–627, 2018.
[37] X. Xu, Q. Li, L. Peng, T.-L. Hsia, C.-J. Huang, and J.-H. Wu. The impact of informational incentives and social influence on consumer behavior during alibaba's online shopping carnival. *Computers in Human Behavior*, 76:245–254, 2017.
[38] C. Yang, H. Yan, D. Yu, Y. Li, and D. M. Chiu. Multi-site user behavior modeling and its application in video recommendation. In *SIGIR*, pages 175–184, 2017.
[39] T. Yuan, J. Cheng, X. Zhang, S. Qiu, H. Lu, et al. Recommendation by mining multiple user behaviors with group sparsity. In *AAAI*, pages 222–228, 2014.
[40] W. Zhang, T. Chen, J. Wang, and Y. Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*, pages 785–788, 2013.
[41] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi. Improving user topic interest profiles by behavior factorization. In *WWW*, pages 1406–1416, 2015.
[42] X. Zheng, N. Liu, and L. Zhao. A study of the effectiveness of online scarce promotion-based on the comparison of planned buying and unplanned buying. In *WHICEB*, page 51, 2013.
[43] M. Zhou, Z. Ding, J. Tang, and D. Yin. Micro behaviors: A new perspective in e-commerce recommender systems. In *WSDM*, pages 727–735, 2018.

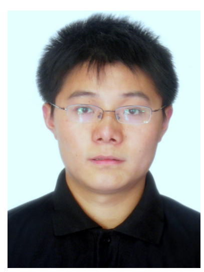
**Jingtao Ding** received the B.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Tsinghua University. His research interests include mobile computing, mobile data mining and user behavior modeling.

**Guanghui Yu** is an undergraduate student in electronic engineering department of Tsinghua University, Beijing, China. His research interests include user behavior modelling.


**Xiangnan He** is currently a professor with the University of Science and Technology of China (USTC). He received his Ph.D. in Computer Science from National University of Singapore (NUS) in 2016, and did postdoctoral research in NUS until 2018. His research interests span information retrieval, data mining, and multi-media analytics. He has over 50 publications appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE, TOIS, and TMM. His work on recommender systems has received the Best Paper Award Honourable Mention in WWW 2018 and ACM SIGIR 2016. Moreover, he has served as the PC member for several top conferences including SIGIR, WWW, MM, KDD etc., and the regular reviewer for journals including TKDE, TOIS, TMM, TNNLS etc.



**Fuli Feng** is a Ph.D. student in the School of Computing, National University of Singapore. He received the B.E. degree in School of Computer Science and Engineering from Baihang University, Beijing, in 2015. His research interests include information retrieval, data mining, and multi-media processing. He has over 10 publications appeared in several top conferences such as SIGIR, WWW, and MM. His work on Bayesian Personalized Ranking has received the Best Poster Award of WWW 2018. Moreover, he has been served as the PC member and external reviewer for several top conferences including SIGIR, ACL, KDD, IJCAI, AAAI, WSDM etc.



**Yong Li** (M'2009-SM'2016) received the B.S. and Ph.D degree in Huazhong University of Science and Technology and Tsinghua University in 2007 and 2012, respectively. During 2012 and 2013, he was a Visiting Research Associate with Telekom Innovation Laboratories and Hong Kong University of Science and Technology respectively. During 2013 to 2014, he was a Visiting Scientist with the University of Miami. He is currently a Faculty Member of the Department of Electronic Engineering, Tsinghua University. His research interests are in the areas of Mobile Computing and Social Networks, Urban Computing and Vehicular Networks, and Network Science and Future Internet. Dr. Li has served as General Chair, Technical Program Committee (TPC) Chair, and TPC Member for several international workshops and conferences. He is currently the Associate Editor of Journal of Communications and Networking and EURASIP Journal of Wireless Communications and Networking

**Depeng Jin** (M'2009) received his B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1995 and 1999 respectively both in electronics engineering. Now he is an associate professor at Tsinghua University and vice chair of Department of Electronic Engineering. Dr. Jin was awarded National Scientific and Technological Innovation Prize (Second Class) in 2002. His research fields include telecommunications, high-speed networks, ASIC design and future internet architecture.